UT-BATTELLE

AMES LABORATORY

ARGONNE NATIONAL LABORATORY

Los Alamos

Lawrence Livermore National Laboratory

NASA

NCAR

Pacific Northwest National Laboratory

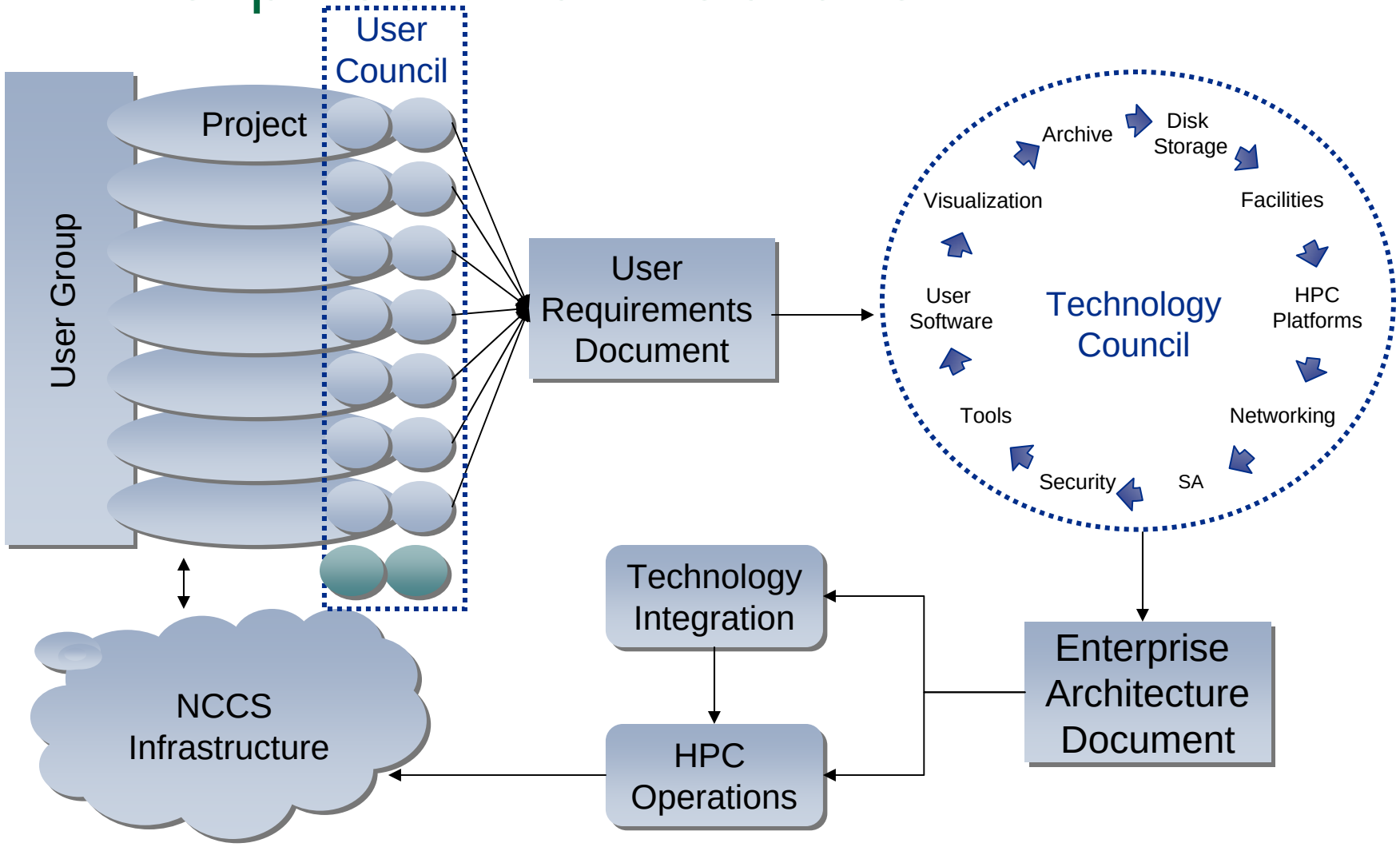
PPPL

Sandia National Laboratories

NCCS User Council

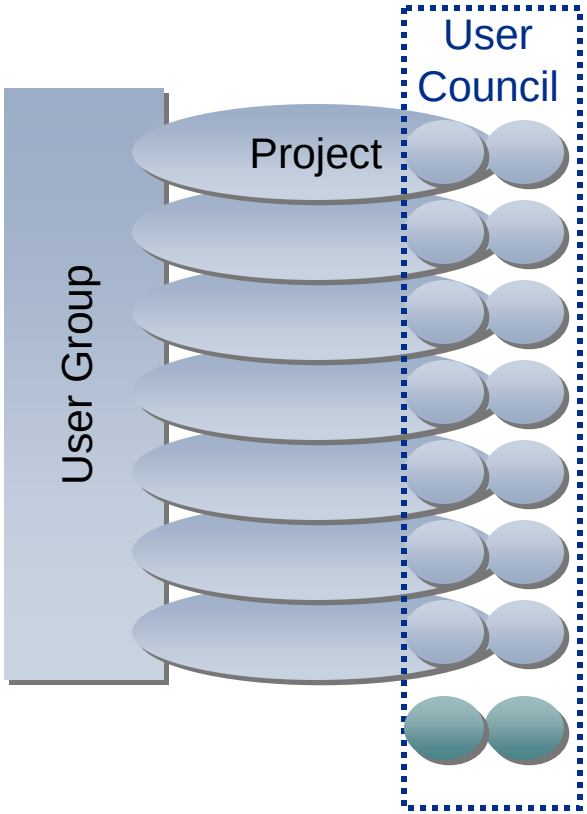
Douglas B. Kothe
Director of Science
National Center for Computational Sciences
Oak Ridge National Laboratory
kothe@ornl.gov

NCCS – From User Requirements to an Enterprise Architecture



NCCS – User Council

Meets directly after allocations are awarded and frequently to adjust document.



An organizational model that will deliver:

- **Strong engagement of research community in setting research and operational requirements**
- **Direct points of contact into each NLCF grand challenge project.**
- **Focused on "future" requirements**
- **Two additional members appointed by NCCS Advisory and NCCS Director to cover emerging application areas**

Each NLCF Grand Challenge project will have one member on the User Council along with that projects liaison from the NLCF Scientific Computing team. These two individuals will ensure the user requirements document is up to date with the requirements from that project.

User Council – Feb 06

- **Membership**
  - **One rep per LCF/INCITE project team and its associated NCCS liaison**
  - **Co-chaired by NCCS (Doug Kothe) & User Group Chair (Jerry Bernholc)**
- **Communication**
  - **Monthly telecon? Individual phone calls, emails, face-to-face mtgs**
  - **This will not be as free form as the Users Group (look at future)**
  - **Co-chairs will set agenda**
- **Product**
  - **Communication and advocacy of our mutual goals and needs**
  - **Annual NCCS requirements doc ("Green Book")**
  - **Influence next generation systems before delivery during the design phase**

User Council Charter

- **Apps-driven requirements for future systems**
  - **Constrained (driven by what will be coming) or**
  - **Unconstrained (what you would like to have achieve science breakthrough)**
  - **Try to look at the "hows" in addition to the "whats"**
  - **Start by gathering metrics, diving down into current state**
- **Track emerging developments in your field**
- **Participate in and/or influence advanced architecture evaluation & design**
- **Be aware of emerging fields ripe for large-scale computational science that can benefit from our efforts**

Future Requirements

- **Consider all aspects of the end-to-end solution: before, after, & underneath the App**
  - **Hardware**
    - **“I need about 100 kB/unknown of on-node memory ”**
    - **“I need 0.01 µs/cell/cycle on-node performance”**
    - **“I need 1000 nearest-neighbor gathers per every unknown for every integration time step”**
  - **Software (everything below the App)**
    - **“The Linux kernel must support dynamic libraries”**
    - **“I need OpenMPI”**
  - **Methods/Models**
    - **“My code currently won’t scale past 500 PEs”**
    - **“I need a DNS-enlightened closure model”**
  - **Pre/Post-Processing**
    - **“I need a parallel Client-Server viz tool”**
    - **“I am going to generate 5PB of data I need around for 5 years”**
    - **“I need <1 month end-to-end turnaround time to satisfy sponsors”**

Future Requirements

- **Future requirements are best articulated and predicted by understanding the footprint of your application on a given platform**
  - **On-node memory, CPU needs**
  - **Off-node local/global communication**
- **In a normalized sense, this footprint is surprisingly constant.**
- **At least this has been my experience**